Loan Approval Classification - Starter Report

Dataset: train.csv (Loan Prediction style dataset). Target: Loan_Status (Y/N).

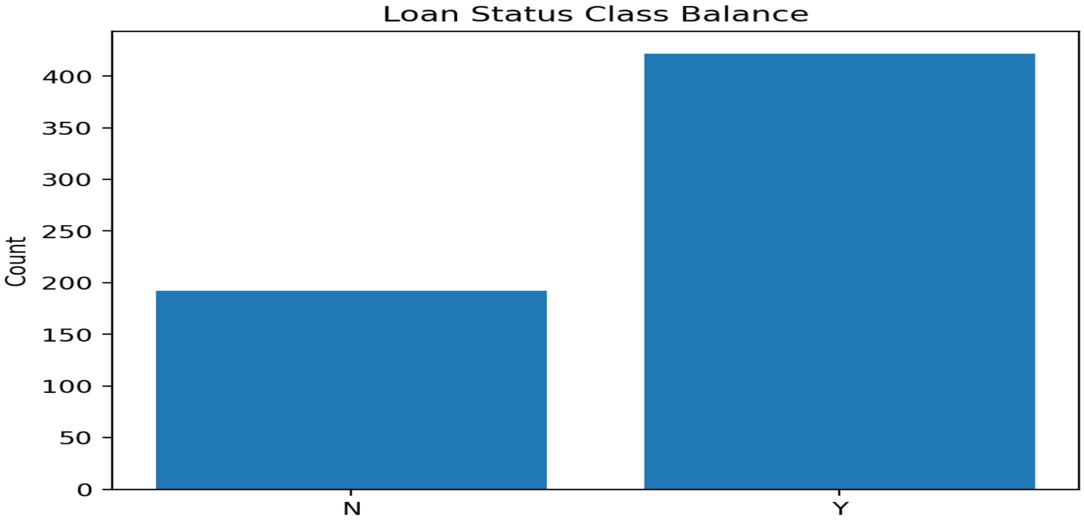
Important: This document is a starter template with computed results. Rewrite the narrative in your own words before submitting.

1. Objective

Goal (choose one): **prediction** of loan approval probability to support faster screening, or **interpretation** to understand the main drivers of approval. For this starter run, the recommended final model is Logistic Regression because it balances performance and explainability.

2. Data Description

Rows: 614, Columns (raw): 13. Identifier column **Loan_ID** was excluded from modeling to avoid leakage. Features include applicant income, loan amount, credit history, and demographics.



Class balance: Approved (Y) = 422 (68.7%), Not approved (N) = 192 (31.3%).

3. Data Cleaning and Feature Engineering

Actions applied in code:

- Dropped identifier column (Loan_ID).

- Imputed missing numeric values with median; missing categorical values with most frequent.

- One-hot encoded categorical variables.

- Standard scaled numeric variables (needed for KNN/SVM/regularized linear models).

4. Models Trained and Evaluation

Train/test split: 80/20 stratified. Models compared: Logistic Regression, KNN, Decision Tree, RBF SVM. Metrics reported: Accuracy, F1, Jaccard, LogLoss.

Model	Accuracy	F1	Jaccard	LogLoss
LogisticRegression	0.862	0.908	0.832	0.390

KNN	0.854	0.903	0.824	0.388
DecisionTree	0.821	0.879	0.784	2.138
SVM_RBF	0.854	0.903	0.824	0.415

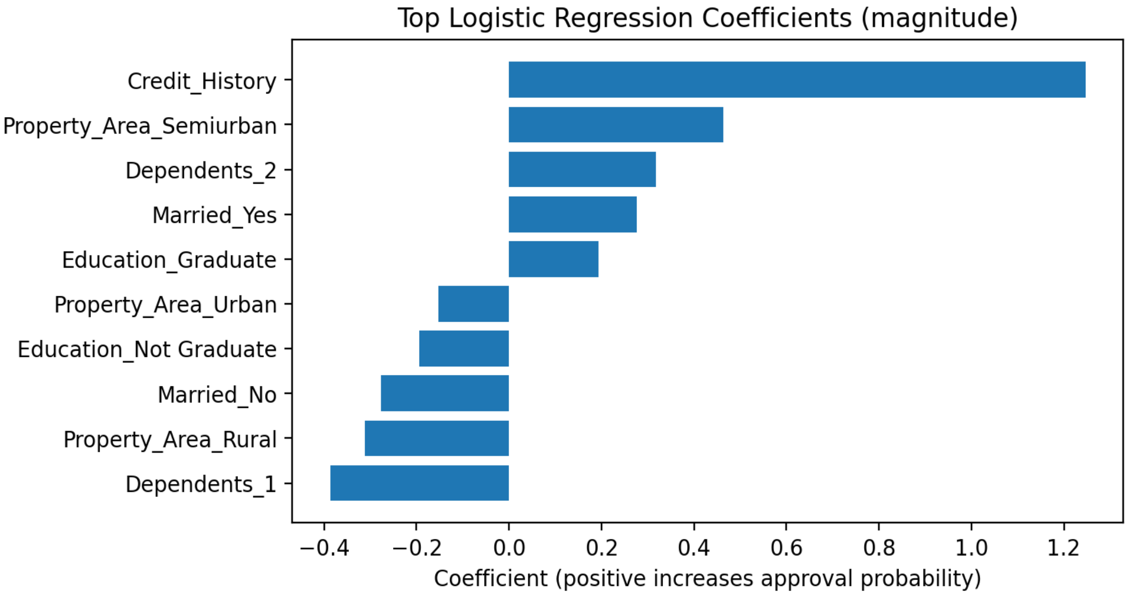
5. Recommended Model

Recommended: **Logistic Regression**. It achieved the best overall F1/Accuracy in this run and provides interpretable coefficients for explaining drivers to stakeholders.

6. Key Findings (from Logistic Regression)

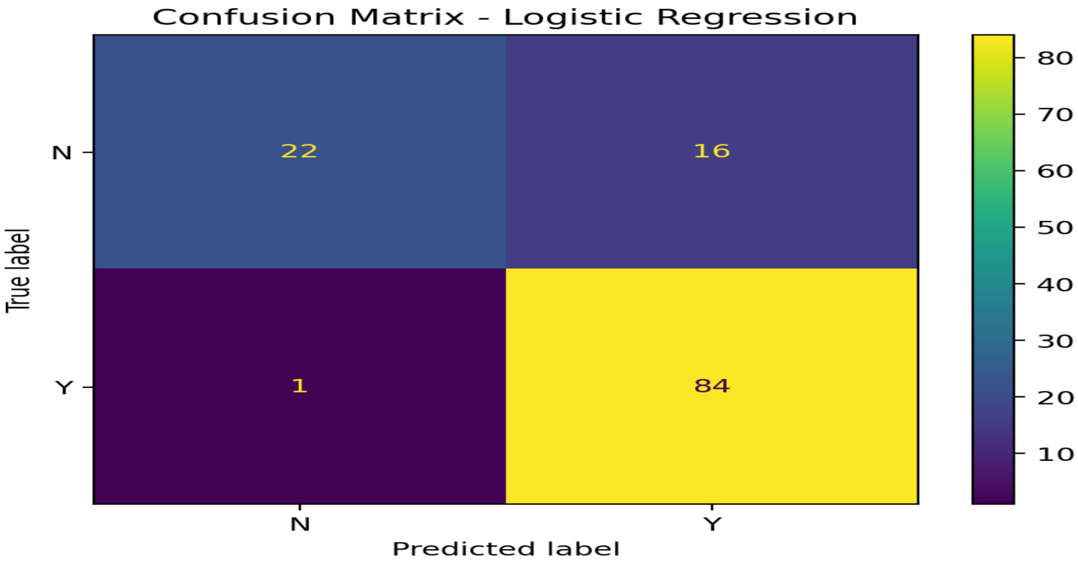
Top drivers (by coefficient magnitude) from the fitted model:

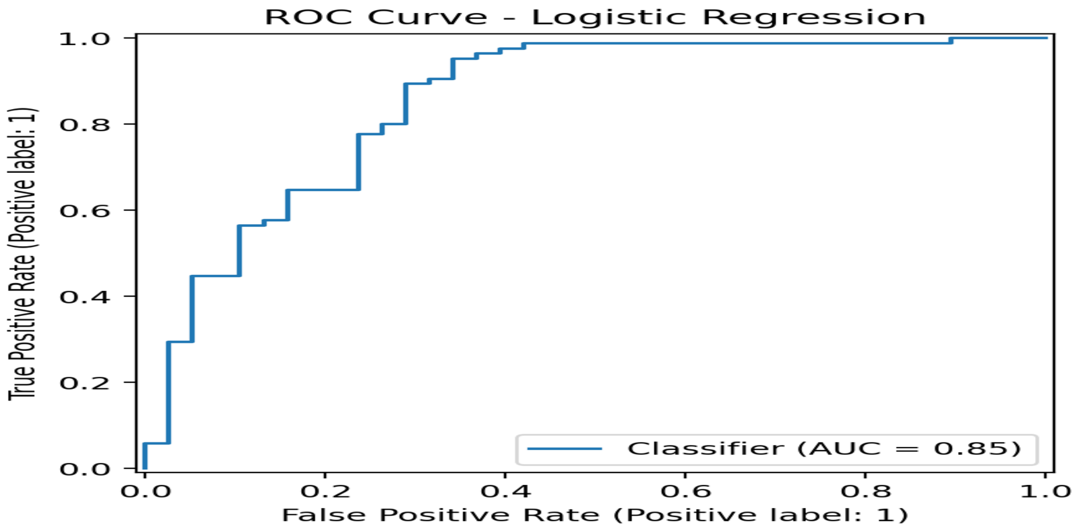
- **Credit_History**: coef +1.247
- **Property_Area_Semiurban**: coef +0.464
- **Dependents_1**: coef -0.387
- **Dependents_2**: coef +0.318
- **Property_Area_Rural**: coef -0.311
- **Married_No**: coef -0.277

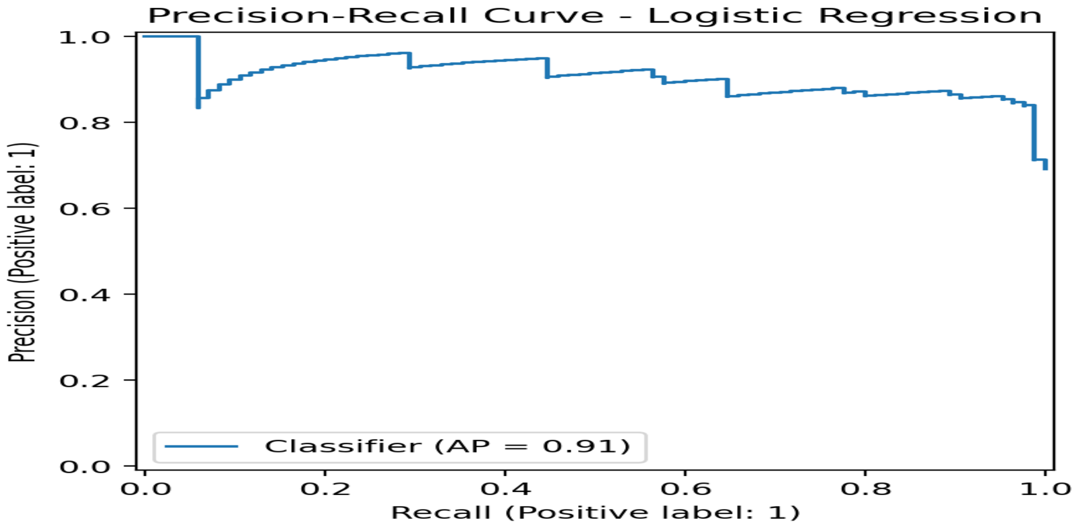


7. Performance Details on Test Set (Logistic Regression)

Accuracy: 0.862 | Precision: 0.840 | Recall: 0.988 | F1: 0.908


Confusion Matrix - Logistic Regression
N
Y
True label
22
16
1
84
N
Y
Predicted label
80
70
60
50
40
30
20
10


ROC Curve - Logistic Regression
True Positive Rate (Positive label: 1)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: 1)
Classifier (AUC = 0.85)


Precision-Recall Curve - Logistic Regression
Precision (Positive label: 1)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Recall (Positive label: 1)
Classifier (AP = 0.91)

8. Limitations and Next Steps

Limitations to mention (edit as needed):

- Class imbalance (more approvals than denials) means accuracy can look strong even if the model produces many false positives.

- Feature set is limited; important underwriting signals (debt-to-income, employment length, credit score details) are missing.

- This is a single random split; results should be confirmed with stratified cross-validation and threshold tuning.

Next steps you can propose:

- Tune decision threshold to reduce false positives if the business cost of approving risky loans is high.

- Try Gradient Boosting (XGBoost/LightGBM or sklearn HistGradientBoosting) for potential performance gains.

- Add calibration (Platt/Isotonic) if probabilities are used for decisioning.

- Re-run with additional features and monitor drift over time.